WHAT IS CLAIMED IS:

1. A method for loading software on a plurality of processors in a heterogeneous processor environment, said method comprising:
retrieving a file using a first processor;

detecting a processor identifier that corresponds to the file;

determining whether to load the file on a second processor based upon the processor identifier; and

loading the file onto the second processor in response to the determination.

2. The method as described in claim 1 further comprising:
executing a program on the first processor;

loading a runtime loader onto the first processor in response the execution; and

performing the retrieving, detecting, and the determining using the runtime loader.

3. The method as described in claim 1 wherein the file is an executable file.

4. The method as described in claim 3 further comprising:
sending a plug-in to the second processor using the first processor, the plug-in corresponding to the file;

sending data to the second processor using the first processor, the data corresponding to the plug-in; and processing the data with the plug-in using the second processor.

5. The method as described in claim 3 further comprising: retrieving a plug-in using the second processor, the plug-in corresponding to the file;

retrieving data using the second processor, the data corresponding to the plug-in; and

processing the data with the plug-in using the second processor.

6. The method as described in claim 3 wherein the executable file is in a file format, and wherein the file format is selected from the group consisting of an ELF format, an XCOFF format, and a PECOFF format.

7. The method as described in claim 1 wherein the processor identifier is a machine type, the determining further comprising:

extracting the machine type from the file; and

comparing the machine type to a plurality of machine types.

8. The method as described in claim 1 wherein the file is part of a combined file, and wherein the processor type corresponds to one or more section headers from a plurality of section headers.

9. The method as described in claim 1 wherein the file is part of a combined file, and wherein the combined file

includes one or more processor identifiers that correspond to the first processor.

10. The method as described in claim 1 wherein the first processor is a processing unit and wherein the second processor is a synergistic processing unit.

11. An information handling system comprising:

a plurality of processors in a heterogeneous processor environment;

a memory accessible by the plurality of processors;

one or more nonvolatile storage devices accessible by the plurality of processors; and

a software loading tool for loading software on a plurality of processors, the software loading tool comprising software code effective to:

retrieve a file using a first processor from one of the nonvolatile storage devices;

detect a processor identifier using the first processor that corresponds to the file;

determine whether to load the file on a second processor based upon the processor identifier; and

load the file onto the second processor in response to the determination.

12. The information handling system as described in claim 11 wherein the software code is further effective to:

execute a program on the first processor;

load a runtime loader onto the first processor in response the execution; and

perform the retrieving, detecting, and the determining using the runtime loader located on the first processor.

13. The information handling system as described in claim 11 wherein the file is an executable file.

14. The information handling system as described in claim 13 wherein the software code is further effective to: send a plug-in to the second processor using the first processor, the plug-in corresponding to the file;

send data to the second processor using the first processor, the data corresponding to the plug-in; and

process the data with the plug-in using the second processor.

15. The information handling system as described in claim 13 wherein the software code is further effective to: retrieve a plug-in using the second processor from one of the nonvolatile storage devices, the plug-in corresponding to the file;

retrieve data using the second processor from one of the nonvolatile storage devices, the data corresponding to the plug-in; and

process the data with the plug-in using the second processor.

16. The information handling system as described in claim 13 wherein the executable file is in a file format, and wherein the file format is selected from the group consisting of an ELF format, an XCOFF format, and a PECOFF format.

17. The information handling system as described in claim 11 wherein the processor identifier is a machine type, and wherein the software code is further effective to: extract the machine type from the file; and

compare the machine type to a plurality of machine types.

18. The information handling system as described in claim 11 wherein the file is part of a combined file, and wherein the processor type corresponds to one or more section headers from a plurality of section headers.

19. The information handling system as described in claim 11 wherein the file is part of a combined file, and wherein the combined file includes one or more processor identifiers that correspond to the first processor.

20. The information handling system as described in claim 11 wherein the first processor is a processing unit and wherein the second processor is a synergistic processing unit.

21. A computer program product stored on a computer operable media for loading software on a plurality of

processors in a heterogeneous processor environment, said computer program product comprising:
means for retrieving a file using a first processor;

means for detecting a processor identifier that corresponds to the file;

means for determining whether to load the file on a second processor based upon the processor identifier; and

means for loading the file onto the second processor in response to the determination.

22. The computer program product as described in claim 21 further comprising:
means for executing a program on the first processor;

means for loading a runtime loader onto the first processor in response the execution; and

means for performing the retrieving, detecting, and the determining using the runtime loader.

23. The computer program product as described in claim 21 wherein the file is an executable file.

24. The computer program product as described in claim 23 further comprising:
means for sending a plug-in to the second processor using the first processor, the plug-in corresponding to the file;

means for sending data to the second processor using the first processor, the data corresponding to the plug-in; and means for processing the data with the plug-in using the second processor.

25. The computer program product as described in claim 23 further comprising:

means for retrieving a plug-in using the second processor, the plug-in corresponding to the file;

means for retrieving data using the second processor, the data corresponding to the plug-in; and

means for processing the data with the plug-in using the second processor.

26. The computer program product as described in claim 23 wherein the executable file is in a file format, and wherein the file format is selected from the group consisting of an ELF format, an XCOFF format, and a PECOFF format.

27. The computer program product as described in claim 21 wherein the processor identifier is a machine type, the means for determining further comprising:

means for extracting the machine type from the file; and

means for comparing the machine type to a plurality of machine types.

28. The computer program product as described in claim 21 wherein the file is part of a combined file, and wherein the processor type corresponds to one or more section headers from a plurality of section headers.

29. The computer program product as described in claim 21 wherein the file is part of a combined file, and wherein the combined file includes one or more processor identifiers that correspond to the first processor.

30. The computer program product as described in claim 21 wherein the first processor is a processing unit and wherein the second processor is a synergistic processing unit.